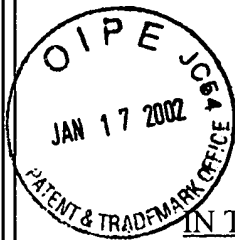


2662



PATENT

Atty. Docket No. 678-636 (P9745)

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICANT: Dong-Jun LEE

SERIAL NO.: 09/973,376

GROUP: Art Unit - Not yet assigned

FILED: October 9, 2001

DATED: November 6, 2001

FOR: DUPLEXING APPARATUS AND  
METHOD IN LARGE SCALE SYSTEM

**RECEIVED**

FEB 0 5 2002

Assistant Commissioner for Patents  
Washington, D.C. 20231

Technology Center 2600

**TRANSMITTAL OF PRIORITY DOCUMENT**

Sir:

Attached is a certified copy of Korean Application No.863/2001 filed on  
January 6, 2001 from which priority is claimed under 35 U.S.C. §119.

Respectfully submitted,

Paul J. Farrell  
Reg. No. 33,494  
Attorney for Applicant

**DILWORTH & BARRESE, LLP**  
333 Earle Ovington Blvd.  
Uniondale, NY 11553  
(516) 228-8484

**CERTIFICATE OF MAILING UNDER 37 C.F.R. §1.8(a)**

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail, postpaid in an envelope addressed to the: Assistant Commissioner for Patents, Washington, D.C. 20231 on November 6, 2001.

Dated: November 6, 2001

  
Paul J. Farrell



대한민국 특허청  
KOREAN INTELLECTUAL  
PROPERTY OFFICE

별첨 사본은 아래 출원의 원본과 동일함을 증명함.

This is to certify that the following application annexed hereto  
is a true copy from the records of the Korean Intellectual  
Property Office.

**CERTIFIED COPY OF  
PRIORITY DOCUMENT**

출원번호 : 특허출원 2001년 제 863 호  
Application Number

출원년월일 : 2001년 01월 06일  
Date of Application

출원인 : 삼성전자 주식회사  
Applicant(s)

RECEIVED

FEB 05 2002

Technology Center 2600



2001 년 03 월 13 일

특 허 청

COMMISSIONER



【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0001
【제출일자】	2001.01.06
【국제특허분류】	H04B
【발명의 명칭】	대형 시스템에서의 이중화 장치 및 방법
【발명의 영문명칭】	DUPLEX APPARATUS AND METHOD OF LARGE SCALE SYSTEM
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이건주
【대리인코드】	9-1998-000339-8
【포괄위임등록번호】	1999-006038-0
【발명자】	
【성명의 국문표기】	이동준
【성명의 영문표기】	LEE, Dong Jun
【주민등록번호】	620815-1804348
【우편번호】	704-360
【주소】	대구광역시 달서구 진천동 진천청구아파트 101동 803호
【국적】	KR
【심사청구】	청구
【취지】	특허법 제42조의 규정에 의한 출원, 특허법 제60조의 규정에 의한 출원심사를 청구합니다. 대리인 이건주 (인)
【수수료】	
【기본출원료】	20 면 29,000 원
【가산출원료】	2 면 2,000 원
【우선권주장료】	0 건 0 원
【심사청구료】	7 항 333,000 원
【합계】	364,000 원
【첨부서류】	1. 요약서·명세서(도면)_1통

**【요약서】****【요약】**

본 발명은, 제1 유니트 및 제2 유니트로 이중화되며, 상기 제1 유니트는 제1 프로세서 및 제1 로컬 메모리를 포함하며, 상기 제2 유니트는 제2 프로세서 및 제2 로컬 메모리를 시스템에서의 이중화 방법에 있어서, 상기 제1 로컬 메모리 및 제2 로컬 메모리를 동시 액세스 및 선택적 액세스를 위한 이중화 채널을 상기 제1유니트 및 제2 유니트 간에 제공하며, 상기 제1 로컬 메모리 및 제2 로컬 메모리를 상기 이중화 채널을 통해 액세스 가능한 이중화 제어 로직부를 제공하는 과정과, 상기 제1 프로세서 및 제2 프로세서중 액티브 프로세서의 메모리 액세스 요구에 의거하여 상기 이중화 제어 로직부가 상기 이중화 채널을 통해 제1 로컬 메모리 및 제2 로컬 메모리를 동시에 액세스하는 과정으로 이루어진다.

**【대표도】**

도 1

**【색인어】**

고장 감내, 이중화, 메모리 동시 액세스

**【명세서】****【발명의 명칭】**

대형 시스템에서의 이중화 장치 및 방법{DUPLEX APPARATUS AND METHOD OF LARGE SCALE SYSTEM}

**【도면의 간단한 설명】**

도 1은 본 발명의 실시 예에 따른 이중화장치 구성도,

도 2는 본 발명의 실시 예에 따른 이중화장치의 구체 블록 구성도,

도 3은 로컬 메모리의 메모리맵 일 예도,

도 4는 본 발명의 실시 예에 따른 이중화를 위한 제어 흐름도.

**【발명의 상세한 설명】****【발명의 목적】****【발명이 속하는 기술분야 및 그 분야의 종래기술】**

<5> 본 발명은 고장 감내 기능 구현을 위한 대형 시스템에서의 이중화 장치 및 방법에 관한 것이다.

<6> 대형 시스템은 고장 감내(fault tolerance)를 실현하기 위해 프로세서 이중화(duplex) 방식을 기본 전제로 채택하고 있다. 프로세서 이중화 방식은 동기식 이중화(synchronized duplex: hot standby) 방식과 비동기식 이중화(asynchronized duplex: warm or cold standby) 방식으로 대별된다. 동기식 이중화 방식은 마이크로 레벨

(micro-level)단위나 명령(instruction)단위 또는 프로그램 수행단위(즉 프로세스 인스턴스(process instance))로 동기를 맞추어 양 쪽 프로세서 모두 동일하게 구동되게 하는 방식이다. 이렇게 양쪽 프로세서들 모두를 동일하게 구동함으로써 상기 동기화 이중화 방식은 장애 발생시 복구를 최대한 빠르게 할 수 있는 이점이 있다. 하지만 동기화 이중화 방식은 양쪽 프로세서들 모두를 구동함에 따른 시스템 부하가 가중되는 단점이 있다. 한편 비동기식 이중화 방식은 두 개의 프로세서들중 어느 한 쪽 프로세서는 실질적으로 동작하지 않고 있다가(standby) 동작중인(active) 프로세서에 장애 발생 시 그 동작을 이어 수행하는 방식이다. 비동기식 이중화 방식은 시스템 부하가 동기식 이중화방식에 비해 상대적으로 적게 걸리지만 시스템 성능에는 단점이 있다. 또한 데이터 일관성 유지 및 장애 발생 시 빠른 재구성(reconfiguration)과 재시동(recovery)을 실현하는 데에 어려움이 있다.

#### 【발명이 이루고자 하는 기술적 과제】

- <7> 고신뢰도를 원하는 대형 시스템에서의 설계 목표는 고도의 고장 감내 기능 구현과 함께 이의 구현에 있어서 성능 저하를 최소로 하고, 동시에 최소한의 소프트웨어 부담으로 데이터 일관성 유지 및 장애 발생 시 빠른 재구성과(reconfiguration) 재시동(recovery)을 실현하는 것이 요망된다. 그리고 시스템내에서 점차 대형화되고 복잡해지는 응용 소프트웨어에서의 부담을 경감시킬 수 있는 이중화 장치 및 방법이 요망된다.
- <8> 따라서 본 발명의 목적은 시스템에 고도의 고장 감내 기능을 구현하면서 그

구현에 있어서 성능 저하를 최소로 하고, 최소한의 소프트웨어 부담으로 데이터 일관성 유지하며 장애 발생시 빠른 재구성(reconfiguration)과 재시동(recovery)을 실현할 수 있는 이중화 장치 및 방법을 제공하는데 있다.

- <9> 본 발명의 다른 목적은 시스템내에서 점차 대형화되고 복잡해지는 응용 소프트웨어에서의 부담을 경감시킬 수 있는 이중화 장치 및 방법을 제공하는데 있다.
- <10> 상기한 목적에 따라, 본 발명은, 본 발명은, 제1 유니트 및 제2 유니트로 이중화되며, 상기 제1 유니트는 제1 프로세서 및 제1 로컬 메모리를 포함하며, 상기 제2 유니트는 제2 프로세서 및 제2 로컬 메모리를 시스템에서의 이중화 방법에 있어서, 상기 제1 로컬 메모리 및 제2 로컬 메모리를 동시 액세스 및 선택적 액세스를 위한 이중화 채널을 상기 제1유니트 및 제2 유니트간에 제공하며, 상기 제1 로컬 메모리 및 제2 로컬 메모리를 상기 이중화 채널을 통해 액세스 가능한 이중화 제어 로직부를 제공하는 과정과, 상기 제1 프로세서 및 제2 프로세서중 액티브 프로세서의 메모리 액세스 요구에 의거하여 상기 이중화 제어 로직부가 상기 이중화 채널을 통해 제1 로컬 메모리 및 제2 로컬 메모리를 동시에 액세스하는 과정으로 이루어짐을 특징으로 한다.

#### 【발명의 구성 및 작용】

- <11> 이하 본 발명의 바람직한 실시 예들을 첨부한 도면을 참조하여 상세히 설명한다. 도면들 중 동일한 구성요소들은 가능한 한 어느 곳에서든지 동일한 부호들로 나타내고 있음에 유의해야 한다. 또한 본 발명의 요지를 불필요하게 흐릴 수 있는 공지 기능 및 구성에 대한 상세한 설명은 생략한다.

<12> 도 1은 본 발명의 실시 예에 따른 이중화장치 구성도로서, 시스템내에는 시스템 신뢰성과 가용성을 보정하기 위해서 동일한 구성인 2개의 MPU(Main Processor Unit)(2a,2b)가 보드형태로서 실장되어 있다. 동일한 구성인 2개의 MPU들(2a,2b)중 하나는 액티브 모드(active mode)가 되고 나머지 하나는 스탠바이 모드(standby mode)가 된다. 2개의 MPU(Main Processor Unit)(2a,2b)는 VME(Versa Module Europa)버스로 연결된다. 도 1에서, IPCU(Inter Processor Communication Unit)(8)는 이중화된 두 개의 MPU(2a,2b)와 타 프로세서간 통신을 위한 인터페이스부로서 본 발명의 요지(aspect)와는 관련성이 없어 더 이상의 상세한 설명은 생략될 것이다.

<13> 본 발명의 실시 예에 따른 이중화장치는 시스템에 고도의 고장 감내 기능을 구현하면서 그 구현에 있어서 성능 저하를 최소로 하고, 최소한의 소프트웨어 부담으로 데이터 일관성 유지하며 시스템 동작중 장애 발생시 빠른 재구성(reconfiguration)과 재시동(recovery)을 실현할 수 있도록 한다. 이를 위해 본 발명의 실시 예에서는 MPU(2a,2b)에 동일한 구성인 이중화 하드웨어부(4a,4b)가 구비되며, MPU(2a)와 MPU(2b)간에는 이중화 채널인 데이터 채널(data channel) D-CH와 제어 채널(control channel) C-CH가 연결된다. 데이터채널 D-CH와 제어채널 C-CH로 구성되는 이중화 채널은 MPU(2a,2b)내에 구비된 로컬 메모리(6a,6b)에 데이터가 동시 라이트될 수 있도록 하기 위해 사용되는 채널이다. 보다 구체적으로 설명하면, 데이터 채널(data channel) D-CH는 MPU(2a)의 로컬메모리(6a) 및 MPU(2b)의 로컬메모리(6b)에 데이터 동시 라이트(Concurrent write)시 그 데이터와 어드레스 및 관련 제어 신호를 전송하기 위해 사용되는 채널이다. 그리고 제어 채널(control channel) C-CH는 이중화된 MPU(2a,2b)간의 액티브/스탠바이 협상(negotiation) 및 정보 교환을 위해 사용되는 채널이다. 또한 상기 제어채널 C-CH는 이





중화 협상 후의 하드웨어적인 제어를 위해 MPU(2a,2b)의 상태(즉 액티브/스탠바이 여부), 이중화 연결 상태 및 이중화 사이클 시작 및 종료 신호를 송수신하기 위한 채널이다.

<14> 도 2는 본 발명의 실시 예에 따른 이중화장치의 구체 블록 구성도로서, 특히 도 1의 MPU(2a)의 구체 블록 구성을 도시하고 있다. MPU(2b)의 구체 블록 구성은 도 2에 도시된 MPU(2a)의 구체 블록 구성과 동일함을 이해하여야한다. 도 2에서는 일 예로 MPU(2a)를 액티브 MPU로, MPU(2b)를 스탠바이 MPU로 가정하고 있다. 두 개의 MPU(2a,2b)는 어느 쪽도 액티브 MPU나 스탠바이 MPU가 될 수 있다. 만약 MPU(2a)가 액티브 MPU가 되면 MPU(2b)는 스탠바이 MPU가 된다. 만약 MPU(2b)가 액티브 MPU가 되면 MPU(2a)는 스탠바이 MPU가 된다.

<15> 도 1의 이중화 하드웨어부(4a)는 도 2에서, 프로세서(10)와 로컬 메모리(6a)를 제외한 나머지 블록들로 구성된다. 즉 이중화 하드웨어부(4a)는 도 2에 도시된 이중화 제어 로직(12), 로컬 제어버퍼(14), 로컬 어드레스버퍼(16), 로컬 데이터 버퍼(18), 이중화 제어 버퍼(20), 이중화 어드레스 버퍼(22), 이중화 데이터버퍼(24)로 구성된다. 그리고 도 1의 데이터채널 D-CH는 도 2의 MPU(2a)의 이중화 제어로직(12)과 MPU(2b)의 이중화제어로직(12)(미도시됨)간에 제어신호, 어드레스, 데이터가 송수신되는 라인들을 통해서 형성되고, 도 1의 제어채널 C-CH는 도 2의 MPU(2a)의 프로세서(10)와 MPU(2b)의 프로세서(10)(미도시됨)간에 제어신호가 송수신되는 라인을 통해서 형성된다.

<16> 도 2에서 액티브 MPU(2a)의 프로세서(10)는 이중화 제어로직(12)을 제어하여, 자신의 로컬 메모리(6a) 및 스탠바이 MPU(2b)의 로컬 메모리(6b)(도 2에는 미도시됨)를 동시에 액세스할 수 있으며, 자신의 로컬 메모리(6a)만을 액세스하거나 아니면 스탠바이

MPU(2b)의 로컬 메모리(6b)만을 액세스할 수 있다. 또한 프로세서(10)는 제어채널 C-CH를 이용해서 이중화된 MPU(2a,2b)간의 액티브/스탠바이 협상(negotiation) 및 정보 교환을 수행한다. 또한 상기 제어채널 C-CH를 이용해서 이중화 협상 후의 하드웨어적인 제어를 위해 MPU(2a,2b)의 상태(즉 액티브/스탠바이 여부), 이중화 연결 상태 및 이중화 사이클 시작 및 종료 신호를 송수신한다.

<17> 이중화 제어 로직(12)은 프로세서(10)의 메모리 액세스 제어에 의거하여 로컬 제어 버퍼(14), 로컬 어드레스 버퍼(16), 로컬 데이터 버퍼(18)와, 이중화 제어버퍼(20), 이중화 어드레스 버퍼(22), 이중화 데이터 버퍼(24) 모두를 인에이블시키거나, 둘중 한 측의 버퍼들만을 인에이블 시킨다. 그래서 자신의 로컬메모리(6a) 및 스탠바이 MPU(2b)의 로컬 메모리(6b)를 동시에 액세스하거나, 자신의 로컬 메모리(6a)만을 액세스하거나, 스탠바이 MPU(2b)의 로컬 메모리(6b)만을 액세스할 수 있다.

<18> 도 3은 로컬 메모리(6a,6b)의 메모리맵 일 예도로서, 사용 메모리 영역의 대응 어드레스 값이 '0x00000000 ~ 0x03ffffff'임을 보여주기 위한 도면이다.

<19> 액티브 프로세서(10)는 자신의 로컬메모리(6a) 및 스탠바이 MPU(2b)의 로컬 메모리(6b)를 동시에 액세스하도록 하기 위해 메모리 액세스를 위한 어드레스값을 '0x40000000 ~ 0x43ffffff'범위에 있는 값으로 이중화 제어 로직(12)에 제공한다. 액티브 프로세서(10)가 스탠바이 MPU(2b)의 로컬 메모리(6b)만을 액세스하도록 할 필요성이 있을 경우 액티브 프로세서(10)는 메모리 액세스를 위한 어드레스값을 '0x80000000 ~ 0x83ffffff'범위에 있는 값으로 이중화 제어 로직(12)에 제공한다. 액티브 프로세서(10)가 자신의 로컬 메모리(6b)만을 액세스하도록 할 필요성이 있을 경우 액티브 프로세서(10)는 메모리 액세스를 위한 어드레스값을

'0x00000000 ~ 0x03ffffff'범위에 있는 값으로 이중화 제어 로직(12)에 제공한다.

<20> 도 4는 본 발명의 실시 예에 따른 이중화를 위한 제어 흐름도로서, 프로세서(10)에 의해서 수행된다.

<21> 이하 도 1 내지 도 4를 참조하여 본 발명의 실시 예에 따른 이중화 동작을 상세히 설명한다. 하기 설명에서는 MPU(2a)는 액티브모드로 동작하고 MPU(2b)는 스탠바이 모드로 동작하는 것으로 가정한다.

<22> 시스템이 파워 온 되면 이중화된 MPU(2a,2b)는 도 4의 100단계 내지 112단계를 수행함에 의해 둘중 하나는 액티브모드로 나머지 하나는 스탠바이모드로 결정되어진다. 보다 구체적으로 설명하면, 시스템이 파워 온 되면 이중화된 MPU(2a,2b)의 프로세서(10) 각각은 제어채널 C-CH를 이용해서 상대 프로세서에 대한 상태응답을 요청한다. 도 2를 참조하여 MPU(2a)의 프로세서(10)가 MPU(2b)의 프로세서(10)로 상태 응답을 요청하는 경로를 일 예로 설명하면, MPU(2a)의 프로세서(10)는 로컬제어버퍼(14), 이중화제어버퍼(20), MPU(2b)의 이중화제어버퍼(20), 로컬제어버퍼(14)를 통해서 MPU(2b)의 프로세서(10)로 상태 응답 요청을 한다.

<23> 그후 프로세서(10)는 도 4의 102단계에서 상대방 프로세서로부터 응답요청이 있는가를 판단하고 만약 응답요청이 없으면 110단계에서 액티브 모드로 설정하고, 그후 112단계에서 액티브 프로세서로서 정상동작을 한다. 이는 상대방 프로세서가 아직 웨이크업(wake up)되지 않았기 때문에 자신이 먼저 액티브 프로세서가 되는 것이다. 하지만 프로세서(10)는 도 4의 102단계의 판단에서 상대방 프로세서로부터 응답 요청이 있으면 상대방 프로세서가 먼저 액티브 모드가 되었기 때문에 104단계로 진행하여 스탠바이 모드



로 설정한다. 그 후 106단계에서 스탠바이 프로세서로서 정상 동작한 다음 108단계에서 액티브모드의 프로세서로 스탠바이 프로세서가 정상임을 알리는 확인신호를 전송한다.

<24> 한편 프로세서(10)가 액티브 프로세서로서 정상동작을 하게 되면 114단계로 진행하여 스탠바이 프로세서로부터 확인신호가 있는가를 판단한다. 액티브 프로세서(10)는 112단계에서 액티브 프로세서로서 정상 동작 후 워치독 타이머(watch\_dog timer)를 운영하고 있으며, 워치독 타이머에 의해 설정된 기간내에 스탠바이 프로세서로부터 확인신호가 없게 되면, 확인신호 수신 체크하는 사이클(cycle)을 강제로 종료한다. 이는 스탠바이 프로세서의 장애 또는 문제가 액티브 프로세서에 영향을 미치는 것을 방지하기 위함이다

<25> 도 4의 114단계의 판단에서 스탠바이 프로세서로부터 확인신호가 수신되지 않으면 116단계에서와 같이 싱글모드(single mode)로 설정하고, 확인신호가 수신되면 122단계에서와 같이 듀얼모드(dual mode)로 설정한다.

<26> 시스템이 싱글모드로 설정이 된 후 액티브 프로세서(10)가 메모리 액세스 해야할 상황이 발생되면 액티브 MPU(2a)의 액티브 프로세서(10)는 이중화 제어로직(12)을 제어하여 자신의 로컬 메모리(6a)를 액세스하게 한다.

<27> 한편 듀얼모드로 설정이 된 후 액티브 프로세서(10)가 메모리 액세스해야할 상황이 발생되면 액티브 MPU(2a)의 액티브 프로세서(10)는 이중화 제어로직(12)을 제어하여 자신의 로컬 메모리(6a) 및 스탠바이 MPU(2b)의 로컬 메모리(6b)를 동시에 액세스하거나, 자신의 로컬 메모리(6a)만을 액세스하거나, 스탠바이 MPU(2b)의 로컬 메모리(6b)를 액세스하게 한다. 상기의 과정은 도 4의 124단계 내지 136단계에서 수행된다.



<28> 듀얼모드 설정된 후 액티브 프로세서(10)가 메모리 액세스하는 동작을 보다 구체적으로 설명하면 하기와 같다. 액티브 프로세서(10)는 도 4의 124단계에서 메모리 액세스해야할 상황이 발생하였는지를 판단하고, 메모리 액세스해야할 상황이 발생하면 도 4의 126단계로 진행하여 노멀 액세스 상황인가를 판단한다. 만약 노멀 액세스 상황이면 자신의 로컬 메모리(6a) 및 스탠바이 MPU(2b)의 로컬 메모리(6b)(도 2에는 미도시됨)를 동시에 액세스할 수 있도록 이중화 제어로직(12)을 제어한다. 이때 액티브 프로세서(10)는 자신의 로컬메모리(6a) 및 스탠바이 MPU(2b)의 로컬 메모리(6b)를 동시에 액세스하도록 하기 위한 어드레스값을 '0x40000000 ~ 0x43ffffff'범위에 있는 값으로 해서 이중화 제어 로직(12)에 제공한다. 이중화 제어로직(12)은 '4x00000000 ~ 0x43ffffff'범위에 있는 어드레스값이 인가되면 데이터 채널 D-CH를 이용해서 액티브 MPU(2a)에 있는 로컬 제어 버퍼(14), 로컬 어드레스 버퍼(16), 로컬 데이터 버퍼(18)와, 이중화 제어버퍼(20), 이중화 어드레스 버퍼(22), 이중화 데이터 버퍼(24), 스탠바이 MPU(2b)에 있는 이중화 제어버퍼(20), 이중화 어드레스 버퍼(22), 이중화 데이터 버퍼(24)를 인에이블 시킨다. 그에 따라 액티브 프로세서(10)는 자신의 로컬메모리(6a) 및 스탠바이 MPU(2b)의 로컬 메모리(6b)를 동시에 액세스할 수 있다.

<29> 상기와 같은 동작에 의해서 액티브 MPU(2a)가 처리한 모든 일들의 결과가 자신의 로컬 메모리(6a)뿐만 아니라 동시에 스탠바이 MPU(2b)로 전달되어 대응 로컬 메모리(6b)에 라이트되므로, 액티브 MPU(2a) 및 스탠바이 MPU(2b)의 로컬 메모리(6a)(6b)는 동일한 데이터를 유지하게 된다. 이는 시스템 운용 중에 액티브 MPU의 동작에 이상이 발생하더라도 시스템의 동작에 지장 없이 현재의 상태를 계속 유지하는 것이 가능하게 한다.

<30> 한편 도 4의 126단계의 판단에서 메모리 액세스해야할 상황이 노멀 액세스 상황이

아니면 도 4의 130단계로 진행하여 스탠바이 프로세서의 로컬 메모리(6b)를 액세스 요구 상황인지를 판단한다. 이러한 액세스 요구 상황을 운용자가 이중화의 시험을 위해서 사용할 수 있다.

- <31> 도 4의 130단계의 판단에서 스탠바이 프로세서의 로컬 메모리(6b)를 액세스 요구 상황이면, 액티브 프로세서(10)는 스탠바이 MPU(2b)의 로컬 메모리(6b)(도 2에는 미도시됨)만을 액세스할 수 있도록 이중화 제어로직(12)을 제어한다. 이때 액티브 프로세서(10)는 스탠바이 MPU(2b)의 로컬 메모리(6b)만을 액세스하도록 하기 위한 어드레스값을 '0x80000000 ~ 0x83ffffff'범위에 있는 값으로 해서 이중화 제어 로직(12)에 제공한다. 이중화 제어로직(12)은 '0x80000000 ~ 0x83ffffff'범위에 있는 어드레스값이 인가되면 데이터 채널 D-CH를 이용해서 액티브 MPU(2a)의 이중화 제어버퍼(20), 이중화 어드레스 버퍼(22), 이중화 데이터 버퍼(24), 스탠바이 MPU(2b)에 있는 이중화 제어버퍼(20), 이중화 어드레스 버퍼(22), 이중화 데이터 버퍼(24)를 인에이블 시킨다. 액티브 MPU(2a) 및 스탠바이 MPU(2b)의 이중화 어드레스버퍼(22)에는 '0x80000000 ~ 0x83ffffff'범위에 있는 어드레스 값이 아니고 '0x00000000 ~ 0x03ffffff'범위에 있는 어드레스값으로 버퍼링된다. 그에 따라 액티브 프로세서(10)는 스탠바이 MPU(2b)의 로컬 메모리(6b)를 액세스할 수 있다. 상기한 바와 같이 액티브 프로세서(10)가 '0x80000000 ~ 0x83ffffff'범위의 어드레스로 액세스하게 되면 액티브 MPU(2a)의 로컬 메모리(6a)에는 전혀 영향을 미치지 않고 이중화 버스를 통해서 스탠바이 MPU(2b)의 로컬 메모리(6b)를 직접 액세스할 수 있다. 이때에는 반드시 액티브 MPU(2a)와 스탠바이 MPU(2b) 사이의 핸드 셰이킹(handshaking)에 의해서 데이터 송수신이 이루어진다. 단, 스탠바이 MPU(2b)의 문제로 인해서 액티브 MPU(2a)가 확인응답을 받지 못한다면 프로세서(10)는 자신의 위치독 타이

머(watch-dog timer)를 구동시켜서 강제적으로 이 상태에서 빠져 나오게 된다. 액티브 프로세서(10)에서는 시스템의 이중화 상태를 관찰하기 위한 이중화 상태 레지스터를 가지고 있는 것이 바람직하다. 프로세서(10)는 상기 이중화 상태 레지스터를 읽어봄으로써 시스템의<sup>1</sup> 상황을 알아 볼 수가 있으며 적절한 대책도 세울 수 있다.

<32> 한편 도 4의 130단계의 판단에서 스탠바이 프로세서의 로컬 메모리(6b)를 액세스 요구 상황이 아니면 액티브 프로세서(10)는 도 4의 134단계로 진행한다. 도 4의 134단계에서 액티브 프로세서(10)는 자신의 로컬 메모리(6a)만을 액세스 요구하는지를 판단한다. 만약 자신의 로컬 메모리(6a)만을 액세스 요구하면 액티브 프로세서(10)는 자신의 로컬 메모리(6a)만을 액세스할 수 있도록 이중화 제어로직(12)을 제어한다. 이때 액티브 프로세서(10)는 자신의 로컬 메모리(6a)만을 액세스하도록 하기 위한 어드레스값을 '0x00000000 ~ 0x03ffffff'범위에 있는 값으로 해서 이중화 제어 로직(12)에 제공한다. 이중화 제어로직(12)은 '0x00000000 ~ 0x03ffffff'범위에 있는 어드레스값이 인가되면 액티브 MPU(2a)의 로컬 제어버퍼(14), 로컬 어드레스 버퍼(16), 로컬 데이터 버퍼(18)를 인에이블 시킨다. 액티브 MPU(2a)의 로컬 어드레스 버퍼(16)에는 '0x00000000 ~ 0x03ffffff'범위에 있는 어드레스값으로 버퍼링된다. 그에 따라 액티브 프로세서(10)는 액티브 MPU(2a)의 로컬 메모리(6a)를 액세스할 수 있다. 상기한 134 단계 내지 136단계를 수행하여 액티브 프로세서(10)가 자신의 로컬 메모리(6a)를 업데이트하게 되면 제어 채널 C-CH를 통하여 스탠바이 MPU(2b)로 복사 요청(duplication request)을 하는 것이 바람직하다. 그에 따라 스탠바이 MPU(2b)로 복사 요청(duplication request)을 접수한 후 자신의 현재 상태를 점검하여(스탠바이 프로세서의 로컬 메모리 액세스와 복사 요청에 의한 로컬 메모리 액세스) 데이터 채널 D-CH의 사용을 제어채널 C-CH를 통해 중재한

다.

- <33> 상술한 본 발명의 설명에서는 구체적인 실시 예에 관해 설명하였으나, 여러 가지 변형이 본 발명의 범위에서 벗어나지 않고 실시할 수 있다. 따라서 본 발명의 범위는 설명된 실시 예에 의하여 정할 것이 아니고 특허청구범위와 특허청구범위의 균등한 것에 의해 정해 져야 한다.

#### 【발명의 효과】

- <34> 상술한 바와 같이 본 발명은 고신뢰도를 원하는 대형 시스템에서 이중화 프로세서의 메모리를 동시에 액세스하는 기능을 하부 구조(하드웨어 및 OS)에서 담당함으로써 점차 대형화되고 복잡해지는 응용 소프트웨어에서의 부담을 경감하는 효과가 있다. 또한 본 발명은 시스템에 고도의 고장 감내 기능을 구현하면서 그 구현에 있어서 성능 저하를 최소로 하고, 최소한의 소프트웨어 부담으로 데이터 일관성 유지하며 장애 발생시 빠른 재구성(reconfiguration)과 재시동(recovery)을 실현할 수 있다.



**【특허청구범위】****【청구항 1】**

제1 유니트 및 제2 유니트로 이중화되며, 상기 제1 유니트는 제1 프로세서 및 제1 로컬 메모리를 포함하며, 상기 제2 유니트는 제2 프로세서 및 제2 로컬 메모리를 포함하는 시스템에서의 이중화 방법에 있어서,

상기 제1 로컬 메모리 및 제2 로컬 메모리를 동시 액세스 및 선택적 액세스를 위한 이중화 채널을 상기 제1 유니트 및 제2 유니트간에 제공하며, 상기 제1 로컬 메모리 및 제2 로컬 메모리를 상기 이중화 채널을 통해 액세스 가능한 이중화 제어 로직부를 제공하는 과정과,

상기 제1 프로세서 및 제2 프로세서중 액티브 프로세서의 메모리 액세스 요구에 의거하여 상기 이중화 제어 로직부가 상기 이중화 채널을 통해 제1 로컬 메모리 및 제2 로컬 메모리를 동시에 액세스하는 과정으로 이루어짐을 특징으로 하는 이중화방법.

**【청구항 2】**

제1항에 있어서, 상기 이중화 채널은

제 1 로컬메모리 및 제2 로컬메모리에 데이터 동시 라이트시 그 데이터와 어드레스 및 관련 제어 신호를 전송하기 위해 사용되는 데이터 채널과,

이중화된 제1,제2 유니트간의 액티브/스탠바이 협상 및 정보 교환을 위해 사용되며, 이중화 협상 후의 하드웨어적인 제어를 위해 제1,제 유니트의 상태, 이중화 연결 상태 및 이중화 사이클 시작 및 종료 신호를 송수신하기 위한 제어채널로 이루어짐을 특징으로

로 하는 이중화방법.

**【청구항 3】**

제1 유니트 및 제2 유니트로 이중화되며, 상기 제1 유니트 및 제2 유니트가 대응 제1 및 제2 로컬 메모리를 포함하고 있는 시스템에서의 이중화 방법에 있어서,

제 1 유니트 및 제2 유니트에 대응 포함된 제1 및 제2 프로세서가 상대 프로세서로 상태응답을 요청하는 과정과,

상기 상대 프로세서로부터 상태응답이 없으면 액티브 프로세서로서 동작하고, 상태응답이 있으면 스탠바이 프로세서로서 동작하며, 상기 스탠바이 프로세서는 액티브 프로세서로 스탠바이 프로세서 정상임을 알리는 확인신호를 전송하는 과정과,

메모리 액세스 요구가 있으면 액티브 프로세서는 상기 확인신호 수신 유무에 따라 상기 제1 로컬 메모리 및 제2 로컬 메모리를 동시 및 선택적으로 액세스하는 과정으로 이루어짐을 특징으로 하는 이중화방법.

**【청구항 4】**

제3항에 있어서, 상기 액티브 프로세서가 스탠바이 프로세서 정상임을 알리는 확인신호를 수신후 메모리 액세스 요구가 있으면 상기 제1 로컬 메모리 및 제2 로컬 메모리를 동시에 액세스함을 특징으로 하는 이중화방법.

**【청구항 5】**

대형 시스템에서의 이중화 장치에 있어서,

동일한 구성으로 이중화된 제1 유니트 및 제2 유니트를 구비하며, 상기 제1 유니트는 제1 프로세서 및 제1 로컬 메모리를 포함하고, 상기 제2 유니트는 제2 프로세서 및 제2 로컬 메모리를 포함하며,

상기 제1 및 제2 유니트간에 상기 제1 로컬 메모리 및 제2 로컬 메모리를 동시 액세스 및 선택적 액세스를 위한 이중화 채널과,

상기 제1 프로세서 및 제2 프로세서중 액티브 프로세서의 메모리 액세스 요구에 의거하여 상기 이중화 채널을 통해 제1 로컬 메모리 및 제2 로컬 메모리를 동시에 액세스 및 선택적으로 액세스하는 이중화 제어 로직부로 구성함을 특징으로 하는 이중화장치.

**【청구항 6】**

제5항에 있어서, 상기 이중화 채널은

상기 제1 로컬메모리 및 제2 로컬메모리에 데이터 동시 라이트시 상기 이중화 제어 로직부에 의해 전송되는 데이터와 어드레스 및 관련 제어 신호의 전송 통로로서 사용되는 데이터 채널과,

이중화된 제1, 제2 유니트간의 액티브/스탠바이 협상 및 정보 교환을 위해 상기 제1 프로세서 및 제2 프로세서에 의해서 사용되며, 이중화 협상 후의 하드웨어적인 제어를 위해 제1, 제2 유니트의 상태, 이중화 연결 상태 및 이중화 사이클 시작 및 종료 신호를

송수신하기 위한 제어채널로 구성함을 특징으로 하는 이중화장치.

【청구항 7】

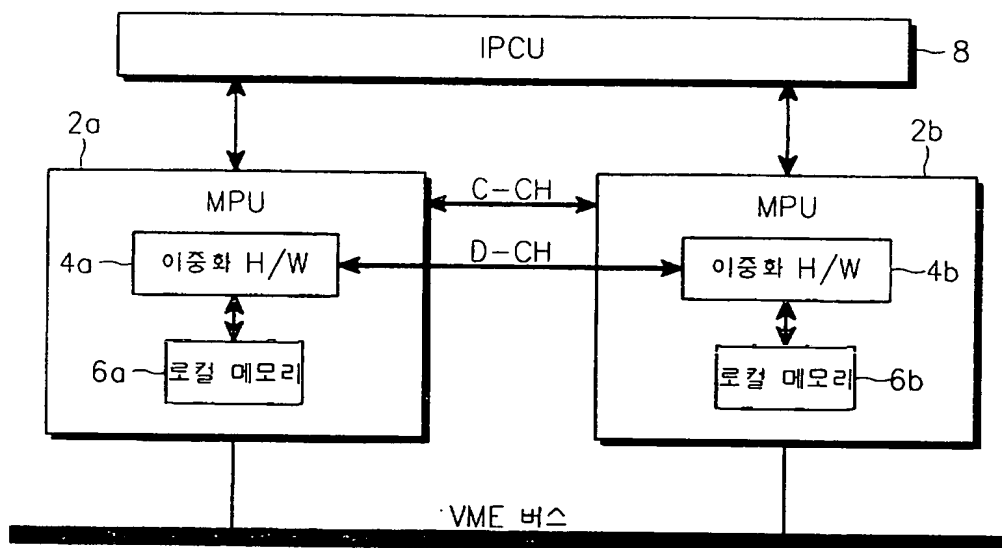
제5항에 있어서,

상기 이중화된 제1 유니트 및 제2 유니트 각각에 구비되며, 자신의 로컬메모리를 액세스하기 위해 사용되는 제어신호, 어드레스, 데이터를 버퍼링하는 로컬 버퍼들과,

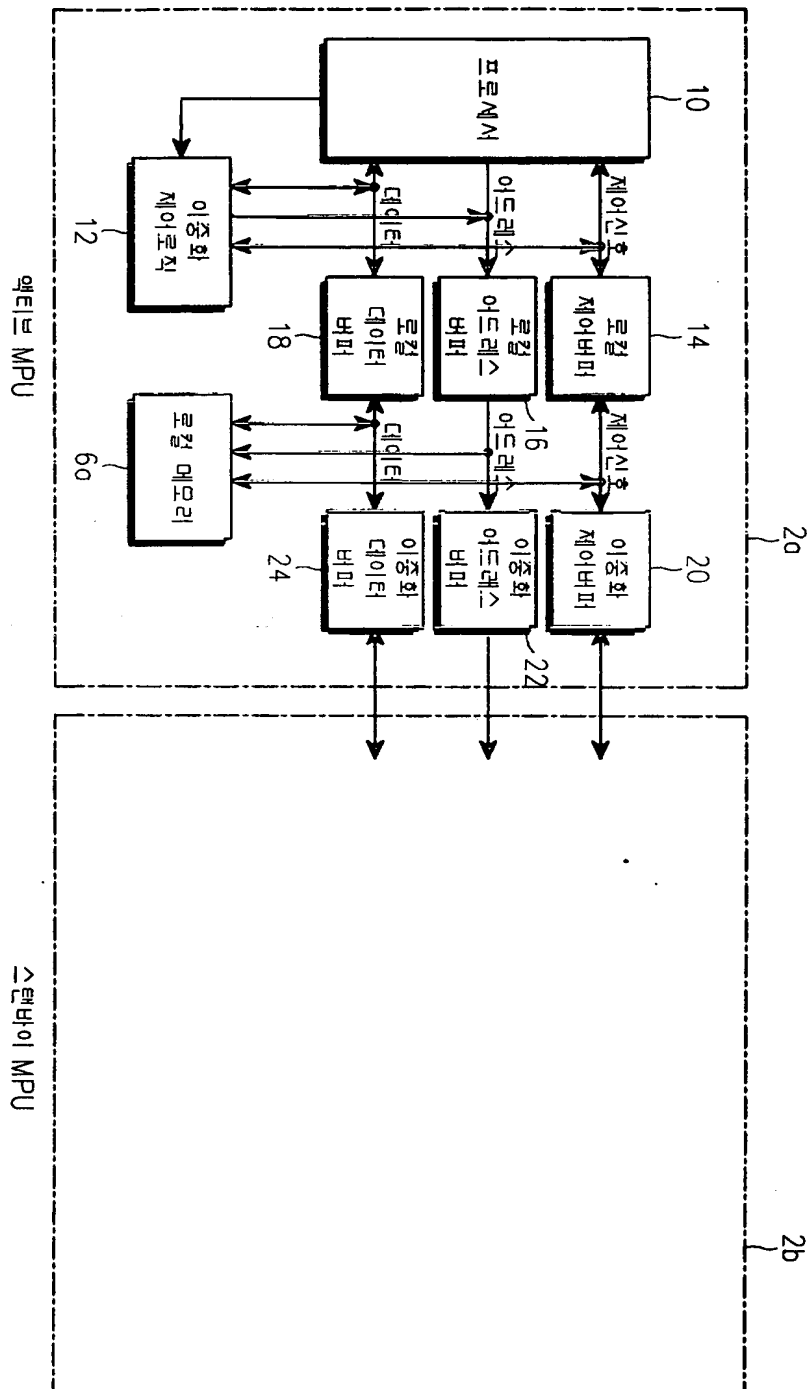
상기 이중화된 제1 유니트 및 제2 유니트 각각에 구비되며, 이중화된 상대의 로컬 메모리를 액세스하기 위해 사용되는 제어신호, 어드레스, 데이터를 버퍼링하는 이중화 버퍼들을 더 구비함을 특징으로 하는 이중화장치.

【도면】

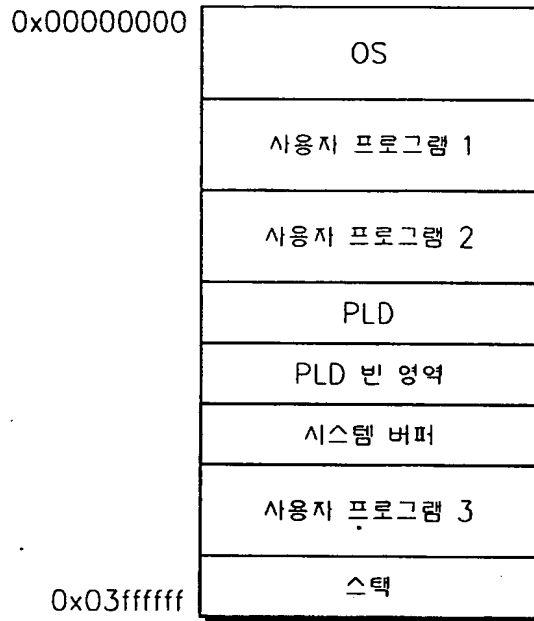
【도 1】



【도 2】



【도 3】



【도 4】

